Automatic Vocal Mixing Using Convolutional Neural Networks

kelian Li
Center for Music Technology
Georgia Institute of Technology
Atlanta, USA
kli421@gatech.edu

Alexander Lerch
Center for Music Technology
Georgia Institute of Technology
Atlanta, USA
alexander.lerch@gatech.edu

***Abstract*—Automatic mixing has become an emerging field in recent years. With many studies attend to create a general-purposed automatic mixing solution, this work presents an approach to perform automatic vocal mixing using convolutional neural networks. Proposed new mixing parameter extraction methods and source separation expand the amount of training data for automatic level balance, compression, equalization, and reverberation. A baseline model is implemented combining data analysis and rule-based approaches. The listening test shows that the baseline model is comparable with the deep learning model. Analysis of the subjective testing result provides insights on the perceptual preferences of music mixing.**

***Index Terms*—automatic mixing, intelligent music production**

I. Introduction

Music mixing is the process of blending multi-track recordings through manipulating audio effects, and the mixing quality is ultimately judged by the preference of listeners. Automatic mixing aims to reach a similar level of quality by computational systems without human interventions [1]. Automatic mixing research can not only increase the efficiency, but also help understand mixing from both scientific and artistic perspectives which may lead to innovations in music production.

This work intends to tackle the problem of vocal and backing track mixing by deep learning. Previous research attempts to create general-purposed multi-track mixing systems which has not yet succeeded. This study focuses on vocal and backing track mixing which has not been addressed in previous research but with wide applications in professional music production and public music participation.

sDerived from hip-hop culture, a considerable amount of music nowadays is created with pre-made accompaniment tracks [2]. Unlike the traditional song-writing and music production where these two processes go hand in hand, the communication between producers and artists is no longer a necessary process. This new norm of music making expanded out of the hip-hop community and has been adapted by the broader music industry as shown in the success of beat-selling platforms like BeatStars [3]. Automatic vocal mixing can accelerate the mixing process and reduce the listening fatigue of the mixing engineers.

Another potential application is karaoke. Mobile karaoke apps enable users to sing and share their performances on social media [4]. Most karaoke mobile apps have some basic vocal mixing features, but general users are unable to produce very good mixes without much knowledge of music mixing.

II. Related work

Early automatic mixing methods are mostly rule-based expert systems. These systems incorporate some common practices by human mixing engineers into algorithms that can perform similar procedures. Reiss [5] introduces the concept of adaptive audio effects, and the key idea is to modifying audio effect parameters by analyzing the audio features of the input signals. More comprehensive overview of rule-based automatic mixing systems can be found in the work by Moffat and Sandler [1] and De Man et al. [6] [7].

As mentioned in the book chapter by Moffat [8], the major limitation of the data-driven automatic mixing is the data collection of mixing parameters. One solution is to let users create training data [9] [10], and another common solution is the reverse engineering approach [11] [12]. Martínez Ramírez et al. [13] propose a method to extract the mixing parameters for black-box audio effects by gradient approximation. All of these approaches require the raw multi-track recordings and the corresponding professional mix-downs.

End-to-end audio transformation provides an alternative path which bypasses the mixing parameter extraction step. Ramírez et al. [14] demonstrates an end-to-end framework for drum mixing with Wave-U-Net [15]. Based on the previous research on black-box audio effects [16], Ramírez et al extend the end-to-end music mixing to processed multitrack training data. Both systems can achieve mixing results that are indistinguishable from the professional mix according to the listening tests [17]. The most latest work shows that the same method can perform end-to-end music mixing style transfer [18].

One obvious drawback of this end-to-end framework is that it does not allow any further adjustments on of the mixing outputs. Steinmetz et al. [19] incorporate differentiable digital signal processing [20] to extract the mixing parameter data, and thus their system can produce human-readable mixing parameters. However, the implementation is very challenging, and the listening test result indicates that the system cannot match with human audio engineers.

Outside of academia, companies have begun adapting automatic mixing into their products. The list includes but not limited to AYAIC[1], RoEx[2], Neutron[3], Balancer[4], and Faders[5]. In the task of automatic mastering, Landr[6], eMastered[7], Ozone[8], BandLab[9], and SoundCloud[10] are considered successful for achieving the commercially usable quality.

III. Methodology

In this work, the audio effects applied to the lead vocals are in the sequence of equalization, compression, reverberation and level balance. To allowing the mixing engineer having the same signal flow, we choose the open-sourced audio plugins MultiEQ, OmniCompressor and FdnRverb by by IEM[11]. Pedalboard[12] is used to load the plugins in the Python environment.

A baseline system and a deep learning system are implemented in this work. The baseline system combines both data analysis and rule-based approaches from previous studies. The deep learning regression model predicts either the direct or intermediate mixing parameters directly from the input signal.

A. Data Collection and Post-processing

Both the data analysis method and the deep learning method rely on the dataset. Using music source separation [21], the vocals and the backing tracks are extracted from 21936 clips of 30-second audio in a subset of the Million Song Dataset [22]. This data is used in level balance and compression. For equalization and reverberation, the dataset is the 100 full-length songs from the train set of MUSDB18HQ [23].

1) Level Balance: Relative integrated loudness [24] between the vocal and the backing track is chosen as the loudness measure which can be easily transferred in the gain changes. The separated Million Song Dataset [22] provides the data for the baseline and the deep learning model. To reduce the influence from outliers, the songs with vocal relative loudness outside of [-10dB, 6dB] are removed from the dataset.

2) Compression: EBU loudness range [25] of the vocal signal is the dynamic range measure in this work. The data is also collected from the separated Million Song Dataset [22]. The songs with vocal loudness range outside of [5dB, 30dB] are removed. An iterative process optimizes the vocal loudness range to 16.36dB ± 1dB by modifying the compression threshold and compression ratio. The compression threshold and the compression ratio are initialized to 0dB and 3:1 respectively. For each iteration, the compression threshold is reduced by 3dB, and the compression ratio is increased by 0.1. The process ends if the loudness range is below 16.36dB or the current compression threshold is lower than -35dB. The attack and release time are set to 30ms and 150ms. This process follows the general concept of adaptive audio effects proposed by Reiss [5].

3) Equalization: The purely rule-based baseline equalization described below does not require any data. The deep learning equalization model uses the 100 songs from the train set of MUSDB18HQ [23], but the data collection method is different from level balance and compression. The main idea is to create ”raw” vocals by applying randomized EQ on the processed vocals. 4 out of 9 center frequencies in [63, 125, 250, 500, 1K, 2K, 4K, 8K, 16K] Hz are randomly chosen. Uniform distributed random gain changes between [-15, 15] dB are applied on the processed vocals using the MultiEQ plugin with the Q factor of 1. 73900 ”raw” vocals in total are created from the 100 processed vocal tracks.

4) Reverberation: Blind reverb estimation remains an unsolved problem in public research [26]. In this work, a commercial plugin Chameleon[13] is utilized to directly extract the estimated reverb impulse responses from the wet vocal signal in the 100 songs from the train set of MUSDB18HQ [23]. The next step is to match the reverb parameters with the impulse responses. As suggested in [27] and [28], genetic algorithms can perform impulse response matching. The genetic algorithm Implementation from PyGAD [29] assists the matching between the FdnRverb plugin parameters and each estimated impulse response. The fitness function minimizes the magnitude spectrogram difference between the targeted and the generated impulse responses. In the end, the reverb parameters are approximated from the wet vocal tracks.

B. Baseline System

1) Level Balance: In an analysis of 64 mixes, De Man et al. [30] find that the relative loudness of the vocal track and the full mix loudness is about -2.7dB. From the 21936 clips in the separated Million Song Dataset [22], we discover that the integrated loudness of the vocals is on average 1.77dB lower than the backing tracks. The baseline system sets the relative integrated loudness of -1.77dB between the vocals and the backing tracks.

2) Compression: The average loudness range of the processed vocals in the separated Million Song Dataset [22] is 16.36dB which is then used as the desired vocal loudness range in the baseline system. The iterative post-processing described above sets the compression parameters which allows the final loudness range to be closed to 16.36dB.

3) Equalization: The baseline equalization is achieved by the rule-based frequency masking reduction method proposed by Hafezi and Reiss [31]. Masking is defined as the process by which the audibility of one sound (the maskee) is reduced

[1] AYAIC, https://www.ayaicinc.com/

[2] RoEx, https://www.roexaudio.com

[3] Neutron, https://www.izotope.com/en/products/neutron/features/mix-assistant.html

[4] Balancer, https://www.sonible.com/balancer/

[5] Faders, https://faders.io/

[6] Landr, https://www.landr.com/

[7] eMastered, https://emastered.com/

[8] Ozone, https://www.izotope.com/en/products/ozone/features/master-assistant.html

[9] BandLab, https://www.bandlab.com/mastering

[10] SoundCloud, https://community.soundcloud.com/mastering-on-soundcloud

[11] IEM, https://plugins.iem.at/docs/plugindescriptions/

[12] Pedalboard, https://github.com/spotify/pedalboard

[13] Chameleon, https://www.accentize.com/chameleon/

by the presence of another sound (the masker). Equalizers are commonly used to reduce the spectral masking in mixing. Our implementation follows [31] with some modifications.

In short, the algorithm firstly identifies the essential and nonessential band of the two signals, and then applies peak filters when one essential band is masked and the same frequency band is nonessential in the maskee signal. The vocal and the backing track can both be the masker and the maskee, but the filter is only applied to the vocal. The incoming signal is firstly gated to reduce noise. The center frequencies of 9 octave bands in use are [63, 125, 250, 500, 1K, 2K, 4K, 8K, 16K] Hz. The lowest 32.5Hz in the original implementation is eliminated. The RMS differences of the 9 bands are normalized by subtracting their mean. The top 6 bands of the vocal and the top 2 bands of the backing track are considered essential. Up to 4 frequency bands are selected where the filters are applied. The Q factor is set to 1.

4) Reverberation: The baseline automatic reverberation combines the data analysis and rule-based approaches. As suggested by Moffat et al., the reverb time may be correlated with the tempo [32]. In our modified equation where the tempo is linearly mapped to the reverb time,

$$RT = -\frac{1}{45}T + \frac{40}{9} \tag{1}$$

RT is the reverb time, and T is the tempo.

Other reverb parameters set to mean values extracted from the train set of MUSDB18HQ [23].

C. deep learning System

Four independent convolutional neural network regression models are trained to predict the direct or intermediate mixing parameters for each audio effects from the mel-spectrogram of the vocal and the backing track. The process of converting intermediate parameters to direct parameters is the same as it in the baseline system. The neural network architectures of the four models are the same except the number of the outputs in the last layer.

To avoid the influence between audio effects, the signal is processed by the audio effect before being fed into the next model. The sequence of the four models is the same as the signal chain. These models are trained to minimize the mean-squared-error between the predictions and the ground truth. During the inference, the predictions are averaged over the mel-spectrogram frames of the entire song.

The vocal relative loudness and vocal loudness range are the ground truth of the level balance model and the compression model. The ground truth for the reverb model are the 10 parameters in the reverb plugin, thus this model is a multi-output regression model. Some of these reverb parameters are empirically more important that the others. To reflect such, the loss function of the reverb model is

$$L_{reverb} = 10 \cdot l_{DW} + 5 \cdot l_{RT} + 5 \cdot l_{RS} + l_{rest} \tag{2}$$

where DW is dry-wet ratio, RT is reverb time, RS is room size, and l_{rest} is the mean-squared-error of the rest of the parameters.

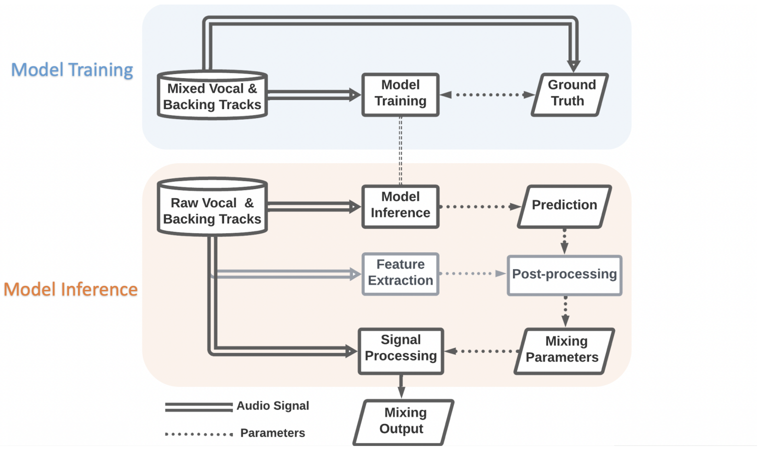

Fig. 1: Overview of the deep learning regression approach

Due to the enormous amount of the training data generated from applying random EQ settings to the vocal tracks, the mel-spectrogram frames of each altered song are concatenated to one frame. Thus, the model output does not need to be averaged because there will be only one set of output.

The ground truth of the EQ model are the gain values in the 9 frequency bands. As mentioned in the data collection section, the peak filters are applied on 4 out 9 center frequencies. This leads to an unbalanced training data where 5 unchanged values. To tackle that, the loss function has to be altered and prevents the model from outputting zeros only. The 9 frequency bands are split into two groups depending on whether the ground truth gain values are changed or not. The modified loss function is

$$L_{EQ} = l_{changed} + 0.1 \cdot l_{unchanged} \tag{3}$$

where $l_{changed}$ is the L2 loss of the 4 frequency bands where the ground truth values are changed, and $l_{unchanged}$ is the loss of the 5 frequency bands where the ground truth values are zeros. During the inference, the top 4 frequency bands which have the largest absolute gain change prediction are selected.

The input feature is the mel-spectrogram of the mono vocal and the backing track at the sample rate of 44100Hz. The FFT size is 2048 with the hop size of 1024. The frames with the vocal integrated loudness below -30dB are removed. The number of mel-filterbanks is 128. Each input feature block includes 64 mel-spectrum temporal steps, and each block represents around 1.51 seconds. The size of the input block is (2, 128, 64). Each input block channel is independently normalized to its maximum value. The neural network architecture is shown below.

Conv2d (input channels = 2, output channels = 8)
BatchNorm
ReLU
MaxPool2d
Conv2d (input channels = 8, output channels = 16)
BatchNorm
ReLU
MaxPool2d
Conv2d (input channels = 16, output channels = 32)
BatchNorm
ReLU
MaxPool2d
Conv2d(input channels = 32, output channels = 64)
BatchNorm
ReLU
MaxPool2d
Dropout (p = 0.3)
MLP (input channels = 768)

For all the convolution layers, the kernel size is 3, the stride is 1, and no zero-padding applied. The kernel size of the max pooling layers is 2.

IV. Experiment

A. Experimental Setups

The ground truth data is normalized between [0., 1.]. The Adam optimizer [33] is used for training. The batch size is 25 except for the EQ model. The learning rate is 1e-3 for the level balance model. In the compression model, the learning rate is 1e-4. In the EQ model, the learning rate is 1e-3 and the batch size is 30. For reverberation, the learning rate is 1e-5 and the weight decay is 1e-5.

The training set for level balance and compression is the 21936 clips of 30-second audio from the source-separated Million Song Dataset [22]. For the EQ and reverb models, the training data is the 100 songs in training set of MUSDB18HQ [23]. The 48 songs in the test set of MUSDB18HQ is used for validation.

B. Objective Evaluation

The validation result in mean-absolute-error is shown in the table below. The mean prediction is what the baseline system uses except for equalization and reverb time. The EQ gain error is the mean absolute error over the 9 frequency bands. DW, RT, FT, RS, LCF, LCG, LCQ, HCF, HCG, and HCQ are the reverb parameters in the FdnVerb plugin and each stands for dry-wet ratio (%), reverb time (s), Fade-in Time (s), Room Size, Low Cut Frequency (Hz), Low Cut Gain (dB), Low Cut Q factor, High Cut Frequency (Hz), High Cut Gain (dB), and High Cut Q factor respectively. Room Size ranges from 1 to 30.

In the tasks of level balance and compression, the CNN model can outperform the baseline system. On reverb parameter prediction, the CNN model can only perform slightly better than the baseline model on most of the parameters. The EQ model successfully converges during the training, but the validation result shows that it lacks the ability of generalization.

C. Listening Test

Due to the subjective nature of mixing, a listening test is conducted to evaluate the mixing quality of the two proposed systems. Besides the mixes by the two systems, the test participants are presented with an additional human mix as the high anchor and a random mix as the low anchor. The human mix uses the same audio plugins from the automatic mixing systems and keeps constant parameters such as the compression attack time and the peak filter's Q factors untouched. Uniform random mixing parameters are created for reverb parameters, compression threshold and ratio, and EQ gains. The vocal relative loudness is randomly generated in the range of [-6, 6] dB in order to make the vocals audible. Other mixing limitations are kept the same way as they are in the human mixes. All mixes are loudness normalized to -28dBFS.

Apart from many previous mixing listening tests, we aim to include more songs in shorter length within the 20-30 minutes test time. Participants are asked to listen to 10 seconds of 15 songs in 4 mix versions. The received feedback indicates that this is a workable method to gather more data to evaluate static mixes. Amateur performances and lower quality recordings are intentionally chosen for this listening test to reflect the wider practice of music making. The 15 songs are selected from DAMP-VSEP [34] which is a real-world dataset of karaoke performances on the Smule app. These songs are in different languages and different music styles. The participants rate the mixing quality in the likert scale of ”very Poor”, ”Poor”, ”Fair”, ”Good”, and ”Very Good”. Two examples of ”Very Good” and ”Very Poor” mixes are given before the test begins. Beyond rating the overall mixing quality, the participants also rate the use of each audio effects. Among the 30 online participants, 24 of them claim to have experiences in audio mixing. 10 participants identify themselves as professional audio engineers.

Figure 2 shows the listening test results in violin plots. For all of the ratings, none of the proposed systems outperforms the human mixes. The baseline system is rated higher than the CNN model in the overall rating with a p-value of $0.13 > 0.05$ in the pair-wise Mann-Whitney U test. It indicates that there is no statistical significance between the two proposed systems. Other pairs have p-values larger than 0.05.

	Level (dB)	loudness range (dB)	EQ gain (dB)	DW	RT	FT	RS	LCF	LCG	LCQ	HCF	HCG	HCQ
CNN	**1.64**	**2.63**	4.48	**6.13**	**1.006**	**0.401**	**7.30**	41.55	**3.62**	**0.355**	**3390**	**5.01**	**0.14**
mean	2.13	2.88	**3.33**	7.16	1.007	0.403	7.31	**40.26**	3.65	0.360	3462	5.35	0.17

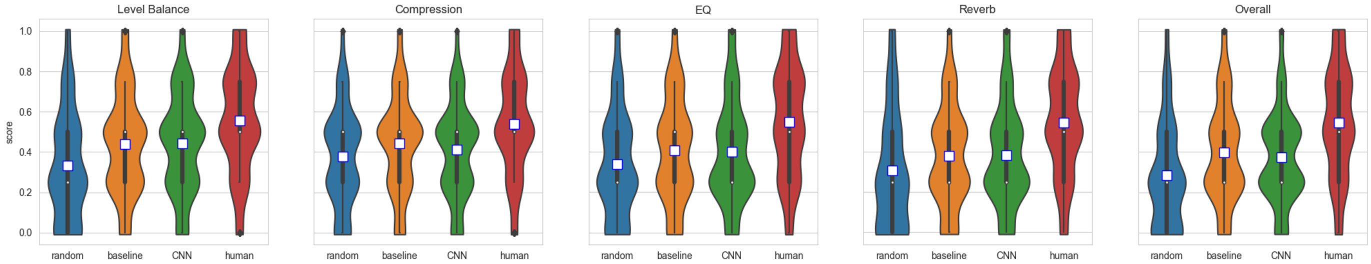

Fig. 2: Listening test results of all participants

V. Discussion

Without analysis on individual songs, the combined result shows that the audio effects are rated similarly to the overall mixing quality with the exception of the compression unit. To further examining this observation, the votes from listeners with different levels of audio engineering skill are show Figure 3 and 4. Audio professionals rate the use of compression drastically higher comparing to the overall rating especially in the random mixes. In contrast, participants with no audio knowledge rate the use of compression similarly to the overall mixing quality. The ratings on compression between audio professionals and inexperienced listeners are also similar. One explanation is that neither audio professionals nor inexperienced listeners can identify the usefulness of compression on vocals in wet mixes. Since the use of compressors is historically linked with non-musical reasons, further investigation on the perception effects of dynamic range compressors is needed.

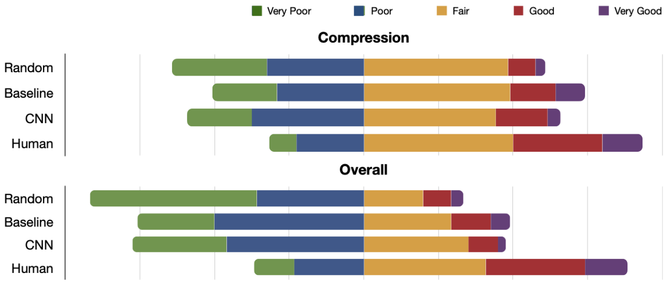

Fig. 3: The compression rating and the overall rating by professional engineers

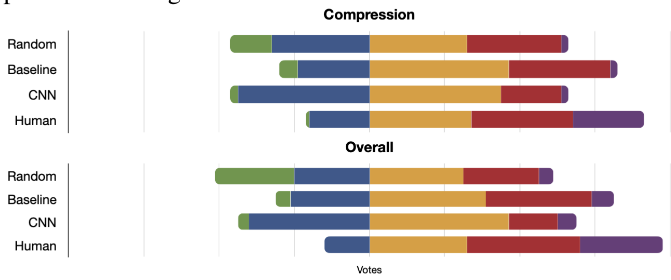

Fig. 4: The compression rating and the overall rating by inexperienced participants

Several participants complain that the performance and the recording quality has influenced their judgements on the mixing quality. They state that it is challenging to differentiate the four mix versions when the quality of the source materials are very good or very poor. This observation aligns with the feedback of the mixing engineer who created the human mixes. Basically, the mixing engineer says that if the recording is in good quality, then the there is not much mixing needed. And if the recording is bad, then mixing cannot help. Obviously this is not a new finding from the experiment, but it raises questions on the effectiveness of mixing on improving the music's perceptual quality in different scenarios. A followup analysis of the source material quality and the participants' ratings can be a preliminary step to identify the true value of mixing.

Furthermore, the connection between the mixing settings and the perpetual mixing quality is largely unexplored in the field of automatic mixing [35] [36] [37]. The lack of knowledge on the tolerance range of mixing parameters and the inherent variations in professional mixes [30] [38] prevents researchers from determining the success of their systems before conducting the listening test. More human perception studies can provide insights on objective metrics and subjective preferences which will eventually facilitate the advancement of automatic mixing. With the available mixing parameters of all 4 mix versions, a future comparison between the distance of mixing settings and the listeners' ratings may contribute to understanding the connection of mixing parameters and mixing perception.

VI. Conclusion

We present an approach to perform automatic vocal mixing using convolutional neural network regression. New mixing parameter extraction methods including and source separation expand expand the amount of training data. The baseline model with combined data analysis and rule-based approaches is proven to be comparable with the deep learning model in the listening test. Further analysis on the subjective test results urges more comprehensive studies to understand the perceptual preferences of music mixing.

VII. Acknowledgments

We would like to appreciate Ben Holst for providing the human mixes for the listening test. We also thank all the participants in the listening test and their valuable feedback.

References

[1] D. Moffat and M. B. Sandler, “Approaches in intelligent music production,” in *Arts*, Multidisciplinary Digital Publishing Institute, vol. 8, 2019.

[2] J. G. Schloss, *Making beats: The art of sample-based hip-hop*. Wesleyan University Press, 2014.

[3] M. Stassen, *Having paid out $150m to creators, BeatStars launches Sony Music Publishing-backed publishing service*, 2021. [Online]. Available: https://www.musicbusinessworldwide.com/having-paid-out-150m-to-creators-beatstars-launches-sony-music-publishing-backed-publishing-service/.

[4] X. Zhou and F. Tarocco, *Karaoke: The global phenomenon*. Reaktion Books, 2013.

[5] J. D. Reiss, “Intelligent systems for mixing multichannel audio,” in *2011 17th International Conference on Digital Signal Processing (DSP)*, IEEE, 2011.

[6] B. De Man, J. Reiss, and R. Stables, “Ten years of automatic mixing,” in *3rd Workshop on Intelligent Music Production*, vol. 15, 2017.

[7] B. De Man, R. Stables, and J. D. Reiss, *Intelligent Music Production*. Audio Engineering Society Presents. Taylor Francis, 2019.

[8] D. Moffat, “Ai music mixing systems,” in *Handbook of Artificial Intelligence for Music*, Springer, 2021, pp. 345–375.

[9] E. T. Chourdakis and J. D. Reiss, “Automatic control of a digital reverberation effect using hybrid models,” in *Dereverberation and Reverberation of Audio, Music, and Speech Conference*, Audio Engineering Society, 2016.

[10] E. T. Chourdakis and J. D. Reiss, “A machine-learning approach to application of intelligent artificial reverberation,” *Journal of the Audio Engineering Society*, vol. 65, no. 1/2, 2017.

[11] D. Barchiesi and J. Reiss, “Reverse engineering of a mix,” *Journal of the Audio Engineering Society*, vol. 58, no. 7/8, 2010.

[12] J. T. Colonel and J. Reiss, “Reverse engineering of a recording mix with differentiable digital signal processing,” *The Journal of the Acoustical Society of America*, vol. 150, no. 1, 2021.

[13] M. A. Martínez Ramírez, O. Wang, P. Smaragdis, and N. J. Bryan, “Differentiable signal processing with black-box audio effects,” in *ICASSP*, IEEE, 2021.

[14] M. Martinez Ramirez, D. Stoller, and D. Moffat, “A deep learning approach to intelligent drum mixing with the wave-u-net,” Audio Engineering Society, 2021.

[15] D. Stoller, S. Ewert, and S. Dixon, “Wave-u-net: A multi-scale neural network for end-to-end audio source separation,” in *Proceedings of ISMIR*, 2018.

[16] M. A. Martınez Ramırez, E. Benetos, and J. D. Reiss, “Deep learning for black-box modeling of audio effects,” *Applied Sciences*, vol. 10, no. 2, p. 638, 2020.

[17] M. A. Martınez-Ramırez, W.-H. Liao, G. Fabbro, S. Uhlich, C. Nagashima, and Y. Mitsufuji, “Automatic music mixing with deep learning and out-of-domain data,” in *Proceedings of ISMIR*, 2022.

[18] J. Koo, M. A. Martinez-Ramirez, W.-H. Liao, S. Uhlich, K. Lee, and Y. Mitsufuji, “Music mixing style transfer: A contrastive learning approach to disentangle audio effects,” *arXiv preprint arXiv:2211.02247*, 2022.

[19] C. J. Steinmetz, J. Pons, S. Pascual, and J. Serrà, “Automatic multitrack mixing with a differentiable mixing console of neural audio effects,” in *ICASSP*, IEEE, 2021.

[20] J. Engel, C. Gu, A. Roberts, *et al.*, “DDSP: Differentiable digital signal processing,” in *International Conference on Learning Representations*, 2019.

[21] A. Défossez, “Hybrid spectrogram and waveform source separation,” in *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, 2021.

[22] T. Bertin-Mahieux, D. Ellis, B. Whitman, and P. Lamere, “The million song dataset,” in *Proceedings of ISMIR*, 2011.

[23] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, *MUSDB18-HQ - an uncompressed version of musdb18*, Dec. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3338373.

[24] EBU–Recommendation, “Loudness normalisation and permitted maximum level of audio signals,” 2011.

[25] E. T. 3342, “Loudness range: A measure to supplement ebu r 128 loudness normalisation,” 2016.

[26] A. Sarroff and R. Michaels, “Blind arbitrary reverb matching,” in *Proceedings of DAFx*, 2020.

[27] A. Cunningham, K. McNally, and P. Driessen, “Automatic impulse response matching for reverb plugins,” in *AES Convention 150*, Audio Engineering Society, 2021.

[28] S. Heise, M. Hlatky, and J. Loviscach, “Automatic adjustment of off-the-shelf reverberation effects,” in *AES Convention 126*, Audio Engineering Society, 2009.

[29] A. F. Gad, *Pygad: An intuitive genetic algorithm python library*, 2021. arXiv: 2106.06158 `[cs.NE]`.

[30] B. Man, B. Leonard, R. King, J. D. Reiss, *et al.*, “An analysis and evaluation of audio features for multitrack music mixtures,” in *Proceedings of ISMIR 2014*, 2014.

[31] S. Hafezi and J. D. Reiss, “Autonomous multitrack equalization based on masking reduction,” *Journal of the Audio Engineering Society*, vol. 63, no. 5, 2015.

[32] D. Moffat and M. Sandler, “An automated approach to the application of reverberation,” in *AES Convention*, 2019.

[33] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2015.

[34] I. Smule, *DAMP-VSEP: Smule Digital Archive of Mobile Performances - Vocal Separation*, Zenodo, Oct. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3553059.

[35] G. Bromham, D. Moffat, M. Barthet, and G. Fazekas, “The impact of compressor ballistics on the perceived

style of music," in *AES Convention 145*, Audio Engineering Society, 2018.

[36] G. Bromham, D. Moffat, D. Sheng, and G. Fazekas, "Measuring audibility threshold levels for attack and release in a dynamic range compressor," in *AES Convention 153*, Audio Engineering Society, 2022.

[37] T. Wilmering, G. Fazekas, and M. B. Sandler, "Audio effect classification based on auditory perceptual attributes," in *AES Convention 135*, Audio Engineering Society, 2013.

[38] A. Wilson and B. Fazenda, "101 mixes: A statistical analysis of mix-variation in a dataset of multi-track music mixes," in *AES Convention 139*, Audio Engineering Society, 2015.